

IN-18  
43069

## Modeling of the Space Station Freedom Data Management System

P31

Marjory J. Johnson

(NASA-CR-188869) MODELING OF THE SPACE  
STATION FREEDOM DATA MANAGEMENT SYSTEM  
(Research Inst. for Advanced Computer  
Science) 31 p

N92-10038

CSCL 22B

Unclassified  
G3/18 0043069

RIACS Technical Report 90.32

August, 1990

Appears in Proceedings of IEEE GLOBECOM '90, December, 1990  
Copyright: IEEE

# **Modeling of the Space Station Freedom Data Management System**

**Marjory J. Johnson**

---

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 311, Columbia, MD 21044, (301)730-2656

---

Work reported herein was supported by the National Aeronautics and Space Administration under Cooperative Agreement NCC 2-387 to the Universities Space Research Association (USRA).

# **Modeling of the Space Station Freedom Data Management System**

*Marjory J. Johnson*

*RIACS*

*NASA Ames Research Center*

*Moffett Field, California 94035*

**Abstract.** The Data Management System (DMS) is the information and communications system on-board Space Station Freedom. Extensive modeling of the DMS is being conducted throughout NASA to aid in the design and development of this vital system. In this paper we discuss activities at NASA Ames Research Center to model the DMS network infrastructure, focusing on modeling of the Fiber Distributed Data Interface (FDDI) token-ring protocol and experimental testbedding of networking aspects of the DMS.

## **1. Introduction**

A dramatic change in space-based communications is occurring, as distributed architectures replace the now-prevalent centralized architectures based on the MIL-STD-1553 command/response multiplex bus [1]. This change is motivated by the need for greater flexibility and enhanced performance to support future space missions, such as Space Station Freedom.

Yet, the introduction of distributed systems into space-mission communications has an associated risk. Inability to control the timing of interactions between distributed elements causes a distributed system to exhibit somewhat unpredictable behavior, unlike the command/response behavior of the MIL-STD-1553 bus. An additional element of risk will be introduced as automation and other artificial-intelligence (AI) technologies are incorporated into space-mission operations.

The use of both distributed systems and AI technologies is being approached conservatively for the Space Station Freedom Data Management System (DMS), because human life will depend on the proper functioning of this system. The DMS will utilize local-area-network technologies, though in the initial phases of Space Station Freedom the DMS (with respect to both hardware and software) is likely to be hierarchical in nature, rather than truly distributed. In addition, the initial use of AI technologies is likely to be limited to a few ground-based support functions. The degree of distributivity of the DMS and the level of use of AI technologies will increase as Space Station Freedom evolves, as mission personnel gain confidence in the use of these technologies. An exciting possibility for advanced evolutionary stages of Space Station Freedom is the management of system operations via the interaction of knowledge-based systems distributed across a network. Such a scenario would significantly reduce the need for human involvement except for emergency backup.

NASA is sponsoring an extensive modeling program to aid in the design of the initial architecture of the DMS and to guide the evolution of the system as distributed-systems and artificial-intelligence technologies assume a more prominent role in the Space Station Freedom program. The purpose of this paper is to discuss activities at NASA Ames Research Center directed towards modeling of the network infrastructure for the DMS. The focus of this work is analysis of the FDDI token ring, which has been baselined for use as the backbone of the DMS.

Network modeling is only one part of an extensive DMS evaluation program that is being conducted at NASA Ames. Other activities that are included in this program include simulation modeling and analysis of the impact of system architecture and work-load distribution on performance of a distributed-processing system, simulation modeling and analysis of dynamic load-balancing strategies for parallel and distributed systems, and experimental testbedding of DMS elements and interfaces to study systems engineering and integration aspects of the DMS. These other activities are not addressed further in this paper.

## 2. Overview of Space Station Freedom

The Space Station Freedom system will be a permanently manned international space facility, a constellation of spacecraft in low-earth orbit with the Space Station Freedom (i.e., the U.S. manned base) as its hub. It is scheduled to be on-orbit in its initial configuration in the mid 1990's. The system will continuously evolve during its expected thirty-year lifetime. In addition to the U.S. manned base, the Space Station Freedom System will include co-orbiting platforms, free flyers, orbital-transfer vehicles, orbital-maneuvering vehicles, and pressurized modules provided by the European Space Agency (ESA) and the National Space Development Agency (NASDA) of Japan. The system will serve as a scientific laboratory, as a manufacturing facility, and as a base for deep-space exploration.

### 2.1. Data Management System

The Data Management System provides information and communication services on-board the manned base; it supports general data-processing and communications functions, as well as command and control functions for the Space Station Freedom. The underlying infrastructure of the DMS is a local area network, with a Fiber Distributed Data Interface (FDDI) token ring as the high-speed backbone. Various subsystems that are connected to the FDDI ring include the Electrical Power System (EPS), the Environmental Control and Life Support System (ECLSS), Guidance, Navigation, and Control (GN&C), and the Thermal Control System (TCS). Some of these systems are single nodes; other systems are subnetworks. Payloads, processors, workstations for crew members, and mass-storage facilities will also be attached to the network.

FDDI is an emerging American National Standards Institute (ANSI) and Organization for International Standardization (ISO) standard for a 100 megabit-per-second fiber-optic token ring. It is a timed-token protocol that supports two classes of service: synchronous service for applications with stringent channel-access requirements, such as packet voice and real-time control, and asynchronous service for applications which do not have such stringent channel-access requirements. While bandwidth is guaranteed for synchronous traffic, asynchronous traffic is transmitted only if the load on the ring is light

enough to support it. The FDDI protocol is complex, but further details are not necessary for this paper. For more detail about the FDDI media-access-control protocol see [3,4,6,9].

## 2.2. End-to-End Communications System

The DMS network is only one component of a wide-area network that reaches from space to ground. This wide-area end-to-end network will be composed of networks having different transmission media, different bandwidths, different error characteristics, etc. In space, the DMS will be connected by gateways to networks within Columbus (the module provided by ESA) and JEM (Japanese Experiment Module). Radio-frequency links connect the DMS, via the Communications and Tracking System, to networks on the various platforms. The ground portion of the end-to-end networking system will be extensive; the end user might be an operator at a ground control center or a scientist located at his home institution, remotely controlling his experiment on-board Space Station Freedom.

The space-to-ground link is provided by the Tracking and Data Relay Satellite System (TDRSS), which currently consists of three communications satellites. Communication protocols for efficient data transmission over the TDRSS links have been recently developed by the Consultative Committee for Space Data Systems (CCSDS) [2].

## 3. LANES Model

Since the FDDI protocol was in the early stages of development when NASA became interested in its use for Space Station Freedom during 1984, the only feasible analysis technique was the use of simulation. Consequently, a group at NASA Ames developed a simulation tool called LANES (Local Area Network Extensible Simulator) to model the FDDI protocol.\* LANES was developed for use by Space Station Freedom contractors, as well as for internal analysis. Hence, portability of the code and generality of the model were primary objectives during development of the simulation. LANES is now available for general use from the Computer Software Management and Information Center (COSMIC), a NASA-software distributor operated through the University of Georgia.

### 3.1. Structure of the Model

LANES is written in SLAM II, a FORTRAN-based simulation language, available through Pritsker & Associates, Inc., West Lafayette, Indiana [8]. The model is structured modularly, in accordance with the Open Systems Interconnection (OSI) network model. Although we originally planned to model all seven layers of the OSI model, we decided instead to combine the upper layers to form a four-layer model. The four layers that are

---

\*LANES also models the Star\*Bus media-access-control protocol, a Carrier Sense Multiple Access with Collision Detection and contention resolution via Time Slot (CSMA/CD/TS) protocol for a fiber-optic star network, developed by Sperry Corporation under contract to NASA Goddard Space Flight Center [11]. One of our first studies using LANES was a comparison of the performance of these two media-access-control protocols. In this paper we discuss the FDDI portion of the model only.

modeled by LANES are called the *physical layer*, *link layer*, *network layer*, and *load layer*. Although the LANES network and load layers model somewhat non-standard functions with respect to the OSI model, delays associated with these layers impact network performance as much as (perhaps more than) the choice of media-access-control protocol.

LANES provides a high-fidelity model of FDDI at the physical and media-access-control layers. The only deviation of our model from the FDDI protocol is that our model restricts each node to generating traffic of at most three priority levels (including both synchronous and asynchronous traffic), whereas the FDDI protocol provides a synchronous traffic class and eight priority levels of asynchronous traffic. The remainder of the model, including part of the link layer, the network layer, and the load layer provide a means for the user to observe some protocol-processing delays that are a significant factor in network performance. Specific features of this nature include a link-layer queue, network-layer buffers, and an absorption delay at the load layer. These mechanisms will be discussed further in Section 3.3. Rather than impose a specific design on implementation aspects of networking, LANES provides sufficient flexibility to enable the user to model and compare the performance of various approaches.

### 3.2. Input to the Simulation

LANES is a parameter-driven simulation. The user specifies input parameters that give a detailed characterization of the network he wishes to model. These parameters are listed in Table 3.1. Since our objective was to develop a general tool for a broad user community, the number of input parameters is large. The user may easily model a non-homogeneous ring, since each node is characterized independently.

There are three categories of parameters: those that describe the physical configuration of the ring (e.g., number of nodes and distance to next node); those that specify FDDI timing parameters (e.g., target-token-rotation time); and those that describe the characteristics of the offered load (e.g., message size).

**Table 3.1. Input Parameters**

Network Parameters	Node Parameters
Number of nodes	Distance to next node
Maximum frame size	Synchronous bandwidth assignment
Size of message header	Buffer capacities
Transmission rate	Message size
Signal speed	Message interarrival time
Internal node delay	Absorption time
Priority threshold	
Target token rotation time	
Percent synchronous traffic	
Backoff time	

Familiarity with networking, in general, and with FDDI, in particular, is necessary to use the simulation meaningfully. Because of the complexity of the FDDI protocol, the user must be careful to specify a consistent set of parameters; the simulation provides no assistance in this regard. Some examples of input combinations that would render a simulation meaningless are:

1. Assigning a value to the target-token-rotation time that is less than the time required to transmit a single frame. This would violate the FDDI timed-token protocol.
2. Generating more synchronous traffic at a single node than it can transmit during its synchronous bandwidth allocation. This would cause synchronous delays at this node to become unbounded, contrary to the guarantees provided by the FDDI protocol with respect to service provided for the synchronous service class.

The user also specifies parameters which control the execution of the simulation, including length of simulated time and time at which to start collecting statistics. It is also possible to collect a detailed trace of all network activities occurring during the simulation. This trace feature is a valuable tool; however, it should be used sparingly because of the large volume of output it produces.

### 3.3. General Traffic Flow

In this section we discuss the general traffic flow modeled by the simulation, from generation of messages at the source to receipt and processing of messages at the destination.

Messages are generated within the load layer of each node. The user specifies message size, message priority, and message interarrival time via appropriate input parameters. LANES models three separate message streams at each node, each with an associated user-specified priority, to enable each node to generate traffic of multiple priorities. Synchronous traffic is represented in the model as highest-priority traffic; multiple priority levels of asynchronous traffic are possible. We felt that provision of three priority levels for each individual node was sufficient to model Space Station Freedom applications, though it would be relatively easy to modify the simulation to model additional asynchronous priority levels.

For each message stream the user specifies message size and message interarrival time (i.e., time between the arrival of successive messages) in terms of statistical distributions. The user may select a constant, exponential, normal, or uniform distribution. In each case the user specifies appropriate characteristic values for the distribution, e.g., the mean of an exponential distribution. If the user selects the constant distribution for message interarrival time (meaning that messages will be generated at constant intervals), the user may specify a value that will offset the generation of the first message. This feature allows the staggering of messages that are generated at different nodes. This has been especially useful in analyzing ring behavior with respect to the synchronous traffic class.

Note that although the user specifies traffic streams in terms of messages, the unit of transmission over the network is a frame. There are no restrictions on the user-specified message size. A message may fit entirely within one frame, or a message may have to be broken down into multiple frames before transmission.

After a message is generated, it is passed from the load layer to the network layer, where messages are segmented into frames and stored in buffers. While the load layer in

LANES has infinite buffer capacity, buffer capacity within the network layer is finite, as specified by the user. Separate buffers may be specified for synchronous and asynchronous messages. A message is passed from the load layer into the network layer only when there is enough room to buffer the entire message. A copy of the message remains in the network-layer buffers until all frames of that message have been received at the destination. Then the space is freed so that another message may be passed from the load layer to the network layer.

The link layer has a finite-capacity queue, as specified by the user, to hold frames which are ready for transmission. The order in which frames are transferred from the network layer to the link layer is highest priority first, and first in, first out within each priority class. Frames are transferred directly from the link-layer queue onto the communications channel according to FDDI transmission rules. The user specifies all pertinent parameters associated with the FDDI protocol, such as target-token-rotation time, synchronous bandwidth, priority threshold, and internal node delay. We have found through experience that it is best to set link-queue capacity to 0, to prevent lower-priority frames from blocking transmission of higher-priority frames.

When frames are received at the destination, they are passed directly to the network-layer receive buffers. The first frame of a message is accepted only if there is sufficient buffer capacity to receive all frames of that entire message. If a frame is rejected, the source node receives a signal from the destination node and enters a backoff period (whose length is specified by the user), during which it will not transmit frames to the node that rejected the frame. Transmission to other nodes on the network continues as normal. Once all frames of a particular message have been received at the destination, the message is ready to be passed to the load layer. Messages are passed sequentially from the network layer to the load layer; the time associated with the transfer of a single message is a user-specified parameter, called absorption time, which represents upper-layer protocol-processing delay at the destination. Once one message is absorbed, another message may begin the process. The order in which messages are absorbed is dependent on message priority.

Although we have not modeled upper-layer protocol processing with any degree of detail, the combination of buffer limitations at the network layer and a non-zero absorption time at the load layer enables the user to observe the impact of delays above the media-access-control level. On the other hand, it is possible to use LANES to model the performance of FDDI as a stand-alone protocol, by setting the absorption time to 0, network-layer buffer capacity so large as to be virtually infinite, and link-queue size to 0. This combination of settings will eliminate any delays that are not caused by the FDDI protocol itself.

### 3.4. Output from the Simulation

Table 3.2 lists the statistics that are available at the end of a simulation run. The comprehensiveness of the output statistics enhances the flexibility of the tool. In this section we explain how to interpret these statistics to analyze network performance.

Output statistics describe both performance of the entire network and performance of each individual node. Each delay statistic listed in Table 3.2 is represented in four ways in the simulation output: average delay, maximum delay, minimum delay, and standard deviation. Load-to-load message delay measures the time from generation of a

message at the source node to the beginning of absorption of the message into the load layer at the destination node. It is composed of several factors: load-to-link frame delay measures queueing delay within the load and network layers at the source node; link-to-link frame delay includes queueing delay within the link layer at the source node, the time to transmit a frame, and the time for the frame to propagate from source to destination; link-to-load message delay measures queueing delay within the destination node, while the message awaits its turn to be absorbed into the load layer. Link-to-load delay, together with absorption time, provides a crude measure of upper-layer protocol-processing delay within the destination node.

**Table 3.2. Output Statistics**

Network Statistics	Node Statistics
Number of token rotations	Total offered load
Total offered load	No. messages delivered
Throughput	No. frames delivered
Utilization	No. frames in network transmit buffer
No. messages delivered	No. frames sent to network transmit buffer
No. frames delivered	No. frames in network receive buffer
Time spent retransmitting rejected frames	No. frames rejected at node
Ave. no. frames in network buffers	No. frames rejected by other nodes
Message size	Time spent retransmitting rejected frames
Load-to-load message delay	Load-to-load message delay
Load-to-link frame delay	Load-to-link frame delay
Link-to-link frame delay	Link-to-link frame delay
Link-to-load message delay	Link-to-load message delay
Wait time for usable token	Wait time for usable token

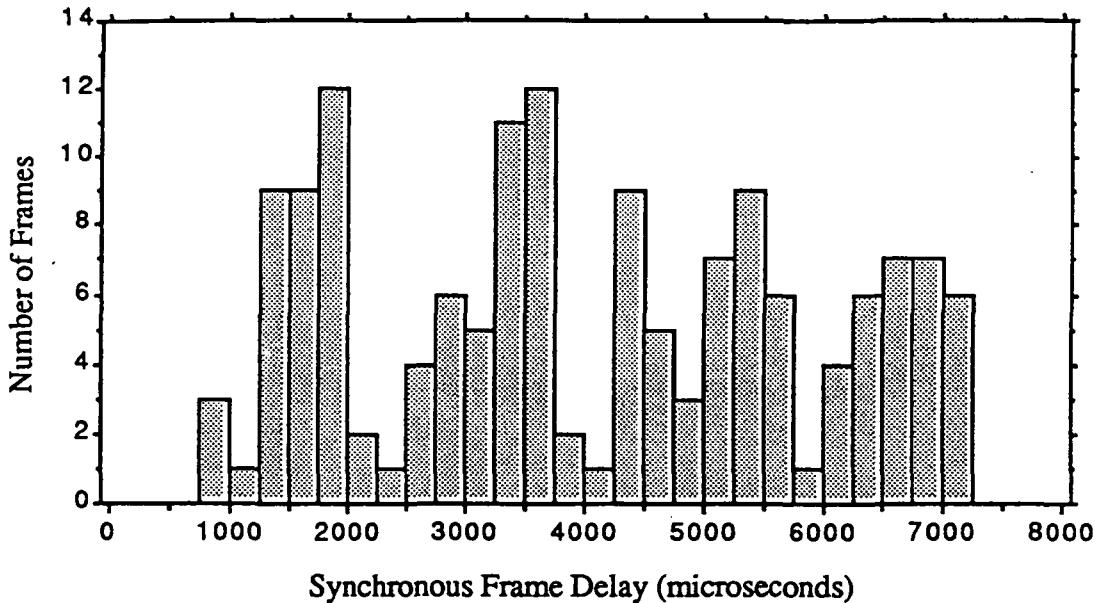
Other statistics include offered load, throughput, utilization,\* protocol-specific statistics (such as number of token rotations, and wait time for usable token), statistics that reveal buffer usage (which are useful to determine if traffic is backing up because of a lack of buffer space), statistics concerning frames that must be retransmitted because there is insufficient buffer space at the destination, and a count of the number of frames and/or messages that are transmitted (tabulated for each node and network wide, according to priority level). Wait time for usable token measures the time that a node which has a frame ready for transmission must wait before a usable token arrives. This statistic is a measure of the responsiveness of the ring. If this time is short, then frames are being serviced almost as soon as they are ready for transmission. If this time is long, it indicates that the ring is heavily loaded.

A valuable source of information about a simulation run, in addition to the output statistics, is the optional trace output, which lists every single activity that occurs on the ring. This can be an extremely useful tool if it is important to see detailed ring activity, though it is tedious to use. One situation where this tool proved useful was to examine ring behavior when the offered load exceeded the capacity of the network. The trace revealed that in this case the network reverted to time-division-multiple-access operation, with each node receiving an equal-sized time slot during each token rotation. Of course, this is the most efficient mode of operation in a heavily loaded ring.

In addition to output available directly from the simulation, we have developed some tools to aid in interpreting results from a simulation run. For example, we have developed a tool to extract all load-to-load message delays for a user-specified node. This tool was used to analyze the impact of various parameter settings on synchronous traffic, since individual message delays are the most important statistic for such a study [5,6]. In addition we have used some personal-computer statistics packages to format simulation output graphically. The combined use of the above tools enabled us to produce the output shown in Figure 3.1.

---

\*Offered load, throughput, and utilization are all expressed as a percentage of the capacity of the network. These values are closely related. Offered load measures the traffic that is generated during simulation run time, exclusive of header overhead. Throughput measures the traffic that was actually delivered, exclusive of header overhead. Throughput and offered load will be approximately the same, unless the offered load exceeds network capacity. Utilization measures network bandwidth that is used during the simulation run. The difference between throughput and utilization reflects the amount of overhead in transmitting the data, including header overhead and possible frame retransmissions. Hence, utilization will generally be larger than throughput.



**Figure 3.1. Frequency Distribution of Synchronous Frame Delays for a User-Specified Node**

### 3.5. Fine Tuning and Validation of the Model

When we first developed the model, it ran so slowly that it was difficult to simulate more than a fraction of a second's network activity. Our next task was to enhance the performance of the simulation itself. We can now run the simulation for several minutes of simulated run time, in an interactive mode. The primary performance-enhancing technique we used is called *token destruction* by Dykeman and Bux [3]. If there are no frames ready for transmission at any of the nodes on the ring, token rotation is halted, thus eliminating all the events associated with token rotation around an empty ring. When a new frame enters the system, a new token is created and placed at the correct point on the ring, based on a calculation using the amount of simulated time that elapsed between token destruction and new-token creation. Although the Dykeman/Bux and the Ames simulations were being developed independently at approximately the same time, both used this mechanism to increase simulation efficiency.

Primary validation of the model was done by hand-checking the results from the trace output. This was a tedious, but effective way to identify and correct errors in the model. Results from our simulation also agreed with the following analytic results:

1. A formula by Ulm [10] which expresses ring utilization as a function of ring latency and average token-rotation time.
2. An equation by Dykeman and Bux [3] for throughput as a function of ring configuration and characteristics of the offered load.

#### 4. Simulation-based Analyses

LANES can be used as a general-purpose tool to analyze the performance of FDDI. Although our use of LANES has been directed towards exploring the capabilities of FDDI with respect to typical Space Station Freedom applications, our results are valid in any setting. The studies that are described in this section illustrate the flexibility of LANES.

##### 4.1. General Performance Analyses

Standard performance analyses of media-access-control protocols typically present graphs of delay versus throughput or utilization versus offered load for various network configurations. Some results of this nature are presented in [6]. LANES provides a user with considerable flexibility to specify many different ring configurations, by assigning different values to such input parameters as number of nodes, distance between nodes, message length, token-rotation time, etc. By varying a single parameter, while holding all others constant, the user can easily determine the impact of this parameter on ring performance.

##### 4.2. Channel Access

With contention-based protocols, an individual node may suffer starvation during periods of ring saturation. Such a situation is unacceptable for the Space Station Freedom DMS. This Space Station requirement motivated an examination of fairness of channel access provided by FDDI. Since synchronous traffic is guaranteed bounded access to the communications channel, we examined access provided for asynchronous traffic.

Using LANES, we modeled a situation in which the offered load exceeded the capacity of the ring, so that every node always has frames ready for transmission over the communications channel. We set the simulation run time to be long, so as to collect meaningful data. The more nodes in the ring configuration, the longer you would need to run the simulation.

The output statistic of interest in this analysis is number of frames transmitted by each node. Statistics measuring delays would be meaningless, because of the infinite queues that build up when the offered load exceeds ring capacity. We first modeled a homogeneous ring, so that no other factors would skew our results. We found that all nodes have equal access to the channel to transmit asynchronous traffic. This result was valid, whether or not synchronous traffic was included as part of the offered load. The test can be repeated for heterogeneous rings. However, the user must be careful to ensure that the situation he is analyzing makes sense. For example, if the user assigns different-sized messages to different nodes (requiring different numbers of frames to transmit the entire message), then the comparison of number of messages transmitted by each node is meaningless.

Results of this study are reported in [6].

#### **4.3. Synchronous Service**

FDDI is typically touted for its 100-megabit-per-second bandwidth capabilities, rather than for the special services it can provide. One such special service is support for synchronous traffic. We conducted a study to explore use of this service on-board Space Station Freedom.

A possible Space Station Freedom application for which the synchronous service class might be appropriate is the transmission of periodic samples from an instrument or laboratory experiment on-board the Space Station. Samples would need to be transmitted regularly, but a reasonable amount of jitter would be tolerable. Based on samples of this type, a scientist on-board Space Station Freedom could control his experiment in real time. We conducted a simulation study using LANES to determine how to set ring parameters to support this type of traffic most efficiently.

In this study, the statistic that is most important is the delay experienced by each individual synchronous frame, rather than average/maximum/minimum/standard-deviation delay statistics provided by LANES. Hence, we modified the simulation to write all frame delays to a file. Then we developed a Pascal program to search this file and select all delays pertaining to a single user-specified node. Finally, we used a personal-computer statistics package to present the information graphically.

Results of this study are reported in [5,6].

#### **4.4. Real-Time Support**

As Space Station Freedom evolves, operations management (including network management) will become increasingly automated. Distributed intelligent agents attached to the DMS will interact autonomously to accomplish everything from routine monitoring; to on-line analysis of system performance; to fault detection, isolation, and reconfiguration. Interactions between these distributed agents must be accomplished in real time, so as to enable the successful handling of emergencies on-board the Space Station Freedom.

We explored the appropriateness of various features of FDDI to handle such real-time fault management via simulation, using LANES. We specified two traffic classes, one for background traffic and another for the real-time traffic to handle emergencies. We conducted a series of simulation runs, varying the volume of background traffic, to determine which of two different techniques for handling the emergency traffic would provide lower delay. The first technique was to assign synchronous bandwidth for the emergency traffic; the second technique was to use asynchronous priorities to distinguish between the two traffic classes, with the emergency traffic having higher priority.

Results of this analysis are reported in [7].

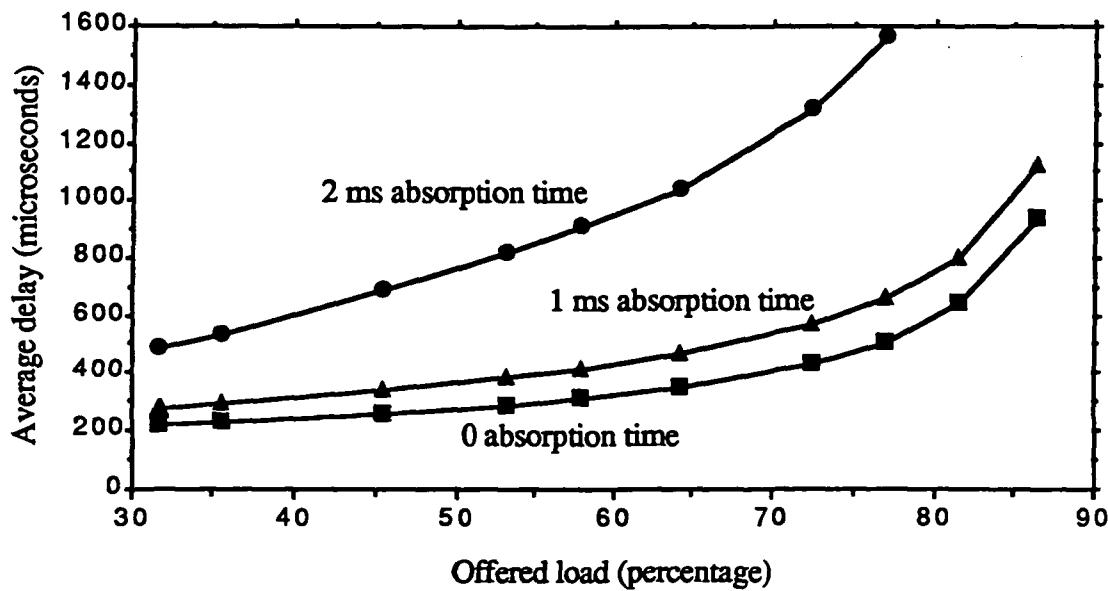
#### **4.5. Effect of Upper-Layer Processing Delays**

It is common knowledge that processing at the upper layers of the OSI network model accounts for a considerable portion of the delay that is observed by the end user of a networking system. End-to-end system performance is of utmost importance for the DMS. Again, simulation is the most accessible way to study this issue early in the design process. Though LANES does not include a detailed model of these upper layers, the user can still observe the effects of upper-layer processing delays on the behavior of an

FDDI ring by limiting buffer capacity at the network layer and by specifying a non-zero value for the absorption-time input parameter. Table 4.1 and Figure 4.1 present results from a simulation study to illustrate the impact of upper-layer processing delays on FDDI performance.

**Table 4.1. Impact of Absorption Time on Bandwidth Utilization**

Offered load (%)	Bandwidth utilization, given a range of absorption times (%)				
	0 ms	1 ms	2 ms	2.5 ms	4 ms
64.2	65.6	65.8	68.9	75.8	99.8
72.5	74.0	74.2	78.4	88.5	--
77.1	78.6	79.0	84.0	94.7	--
81.6	83.3	83.5	89.5	99.2	--
86.4	88.1	88.5	96.4	--	--
92.8	94.6	95.1	99.8	--	--



**Figure 4.1. Effect of Different Absorption Times on Average Load-to-Load Delay**

## 5. Related Testbed Activities

Now that FDDI components can be purchased commercially, we plan to complement our simulation analyses with testbed experimentation. We are currently developing an FDDI testbed at NASA Ames as a development tool for the DMS. In addition to providing a model that faithfully represents the current design of the DMS, we will also use the testbed to determine experimentally how to integrate new computer architectures into the DMS.

Immediate plans include the modeling of fault scenarios on this testbed, to determine how to ensure that the network will respond in a timely fashion to cope with these situations. This is a critical requirement for the DMS, to ensure the safety and reliability of Space Station Freedom. Other objectives include determining how to ensure that the DMS will evolve into a fully distributed system, how to support near-real-time interaction between a ground-based scientist and his on-board experiment, and how to interface with the communications systems in the international modules. Longer-term plans include the extension of our testbed at Ames to an end-to-end testbed including facilities at other sites, to model the entire space-to-ground communications system.

## 6. Conclusion

Major milestones in the Space Station Freedom program are Critical Design Review in 1991, First Element Launch in 1995, Permanent Manned Capability in 1997, and Assembly Complete in 1999. The full DMS will not be on-orbit until 1999. There is a critical need for modeling of the Space Station Freedom Data Management System, both to determine suitability and performance of currently proposed designs and to plan for evolution of the system.

In this paper we discussed DMS network-modeling activities at NASA Ames Research Center. Complementary modeling activities are being conducted at NASA centers throughout the country, as well as at contractor facilities. These studies will have a direct impact on the design of the DMS, both for its initial architecture for 1999 and for its evolutionary development.

**Acknowledgements.** The author wishes to acknowledge the contributions of other members of the group which developed the LANES simulation: Terry Grant, manager, Sharon Doubek, Jim Gibson, Joe Jordan, Keith Richardson, and Larry Webster.

## References

1. "Aircraft Internal Time Division Command Response Multiplex Data Bus," MIL-STD-1553B, Notice 1, February 1980.
2. Consultative Committee for Space Data Systems, *Recommendation for Space Data System Standards. Advanced Orbiting Systems, Networks and Data Links: Architectural Specification*, CCSDS 701.0-B-1, October, 1989.
3. D. Dykeman and W. Bux, "An Investigation of the FDDI Media Access Control Protocol," *Proc. EFOC/LAN '87*, Basel, Switzerland, 1987, IGI Europe - Information Gatekeepers, Inc., Boston, pp. 229-236.
4. *FDDI Token Ring Media Access Control*, American National Standard, X3.139 - 1987.
5. M. J. Johnson, "Analysis of FDDI Synchronous Traffic Delays," *Proceedings of Systems Design and Networks Conference: Putting Local Area Networks to Work*, (IEEE), Santa Clara, CA, April 1988, pp. 65-72.
6. M. J. Johnson, "Performance Analysis of FDDI," *Proc. EFOC/LAN '88*, Amsterdam, The Netherlands, June, 1988, IGI Europe - Information Gatekeepers, Inc., Boston, pp. 285-300.
7. M. J. Johnson, "Performance Issues in Management of the Space Station Information System," in *Integrated Network Management, I*, edited by B. Meandzija and J. Westcott, North Holland, 1989, pp. 423-434.
8. A. Pritsker and C. Pegden, *Introduction to Simulation and SLAM*, John Wiley & Sons, 1979.
9. F. E. Ross, "FDDI - A Tutorial," *IEEE Commun. Mag.*, Vol. 24, No. 5, 1986, pp. 10-17.
10. J. M. Ulm, "A Timed Token Ring Local Area Network and its Performance Characteristics," *Proc. 7th Conf. Local Comput. Networks*, IEEE, Feb. 1982, pp. 50-56.
11. M. Varga, "A Robust 100 MBPS Network for Avionics Application," *Proceedings of the National Aerospace and Electronics Conference (NAE-CON)*, May, 1985.